(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 422 206 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.01.2019 Bulletin 2019/01

(51) Int Cl.:
G06F 17/30 *(2006.01)*

(21) Application number: **17178243.6**

(22) Date of filing: **27.06.2017**

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
MA MD

(71) Applicant: **Deutsche Telekom AG**
53113 Bonn (DE)

(72) Inventors:
- **SCHATZMAYR, Rainer**
  53229 Bonn (DE)
- **AGAPI, Andrei**
  1064CN Amsterdam (NL)

(74) Representative: **Schwöbel, Thilo K. et al**
Kutzenberger Wolff & Partner
Waidmarkt 11
50676 Köln (DE)

(54) METHOD FOR AN ENHANCED MANAGEMENT OF CONFIGURATION INFORMATION AND/OR STATEFUL INFORMATION IN A COMMUNICATION NETWORK, SYSTEM, PROGRAM AND COMPUTER PROGRAM PRODUCT

(57) The invention relates a method for an enhanced management of configuration information and/or stateful information in a communication network, wherein the communication network comprises a plurality of network nodes (11, 12, 13) and a distributed NoSQL database functionality (20), wherein the distributed NoSQL database functionality (20) comprises a plurality of database entities (21, 22, 23), wherein query messages between the plurality of network nodes (11, 12, 13) and the NoSQL database functionality (20) are transmitted using a data modeling language with a graph-like or tree-like data structure, wherein the query messages comprise graphs referencing vertices stored in the plurality of database entities (21, 22, 23), wherein a specific graph of a specific query message comprises references to at least two vertices,
wherein the at least two vertices referenced in the specific graph are distributed within a group of initial graph partitions and/or modified graph partitions across the database entities (21, 22, 23) of the NoSQL database functionality (20).

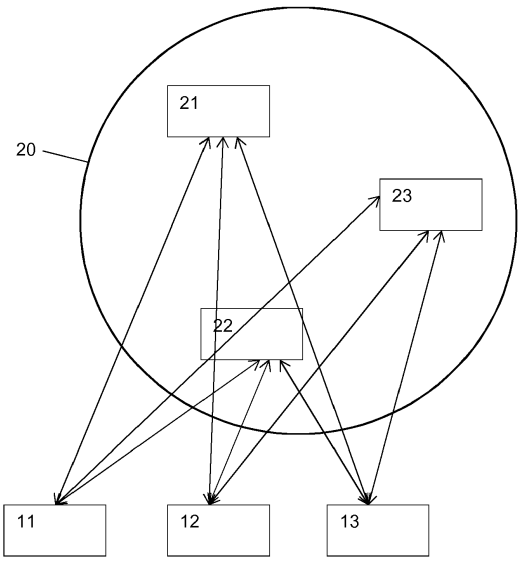

Fig. 1

EP 3 422 206 A1

Description

BACKGROUND

[0001] The present invention relates a method for an enhanced management of configuration information and/or stateful information in a communication network, wherein the communication network comprises a plurality of network nodes and a distributed NoSal database functionality, wherein the distributed NoSal database functionality comprises a plurality of database entities, wherein query messages between the plurality of network nodes and the NoSal database functionality are transmitted using a data modeling language with a graph-like or tree-like data structure, wherein the query messages comprise graphs referencing vertices stored in the plurality of database entities, wherein a specific graph of a specific query message comprises references to at least two vertices.

[0002] Furthermore, the present invention relates to a system for an enhanced management of configuration information and/or stateful information in a communication network, wherein the communication network comprises a plurality of network nodes and a distributed NoSal database functionality, wherein the distributed NoSal database functionality comprises a plurality of database entities, wherein query messages between the plurality of network nodes and the NoSal database functionality are transmittable using a data modeling language with a graph-like or tree-like data structure, wherein the query messages comprise graphs referencing vertices stored in the plurality of database entities, wherein a specific graph of a specific query message comprises references to at least two vertices.

[0003] Additionally, the present invention relates to a program comprising a computer readable program code which, when executed in part on a query profile module, a graph partitioning module, a query router module, a query handler module and/or on a database entity and/or on a network node, causes the query profile module, the graph partitioning module, the query router module, the query handler module and/or the database entity and/or the network node to perform a method according to an embodiment of the present invention.

[0004] Additionally, the present invention relates to a computer program product for an enhanced management of configuration information and/or stateful information in a communication network, the computer program product comprising a computer program stored on a storage medium, the computer program comprising program code which, when executed in part on a query profile module, a graph partitioning module, a query router module, a query handler module and/or on a database entity and/or on a network node, causes the query profile module, the graph partitioning module, the query router module, the query handler module and/or the database entity and/or the network node to perform a method according to an embodiment of the present invention.

[0005] For the management of configuration information and/or stateful information in a communication network, network nodes need to be able to receive configuration data and changes of configuration data from databases as well as write stateful data into the databases/stores. Therein, the handling of query messages is extremely important. A typical, non-limiting example of a data modeling language with a graph-like or tree-like data structure employed for query messages is YANG ("yet another next generation"), which is a data modeling language for the definition of data sent over the NETCONF network configuration protocol. Existing systems have the problem that the storage of vertices referenced/addressed in the query messages is not efficient such that the handling of the query messages involves increased workload.

SUMMARY

[0006] An object of the present invention is to provide an improved method for an enhanced management of configuration information and/or stateful information in a communication network, wherein an improved storage/database functionality is provided such that the required resources for routing and/or answering query messages are minimized.

[0007] The object of the present invention is achieved by a method for an enhanced management of configuration information and/or stateful information in a communication network, wherein the communication network comprises a plurality of network nodes and a distributed NoSal database functionality, wherein the distributed NoSQL database functionality comprises a plurality of database entities, wherein query messages between the plurality of network nodes and the NoSal database functionality are transmitted using a data modeling language with a graph-like or tree-like data structure, wherein the query messages comprise graphs referencing vertices stored in the plurality of database entities, wherein a specific graph of a specific query message comprises references to at least two vertices, wherein the at least two vertices referenced in the specific graph are distributed within a group of initial graph partitions across the database entities of the NoSQL database functionality by the following steps:

-- in a first step, a plurality of query messages is tracked and analyzed by a query profile module, wherein the query profile module generates workload information for a specific vertex referenced in the specific graph of the specific query message of the plurality of query messages associated to storing the specific vertex on a specific database entity out of the plurality of database entities, wherein the workload information for the specific vertex is a function of

-- the number of query messages of the plurality of query messages referencing the specific ver-

tex,

-- the frequency of query messages of the plurality of query messages referencing the specific vertex,

-- wherein, in case the specific vertex is the first or last vertex of a graph partition, the workload information is additionally a function of a communication cost value associated with hopping/transitioning to a vertex in a different graph partition,

-- in a second step - after the first step - in dependence of the workload information, a partitioning of the specific graph into a group of initial graph partitions is performed by a graph partitioning module, wherein each graph partition of the group of initial graph partitions is distributed to at least one different database entity of the plurality of database entities, and/or

wherein the at least two vertices referenced in the specific graph are distributed within a group of modified graph partitions across the database entities of the NoSQL database functionality by the following steps:

-- in a first step, a plurality of query messages is tracked and analyzed by a query profile module, wherein the query profile module generates workload information for a specific vertex referenced in the specific graph of the specific query message of the plurality of query messages associated to storing the specific vertex on a specific database entity out of the plurality of database entities, wherein the workload information for the specific vertex is a function of

-- the number of query messages of the plurality of query messages referencing the specific vertex,

-- the frequency of query messages of the plurality of query messages referencing the specific vertex,

-- wherein, in case the specific vertex is the first or last vertex of a graph partition, the workload information is additionally a function of a communication cost value associated with hopping/transitioning to a vertex in a different graph partition,

-- in a second step - after the first step - in dependence of the workload information, a partitioning of the specific graph into a group of modified graph partitions is performed by a graph partitioning module, wherein each graph partition of the group of modified graph partitions is distributed to at least one different database entity of the plurality of database entities,

wherein a further query message, transmitted from a specific network node of the plurality of network nodes and directed either

-- at editing and/or writing and/or retrieving stateful information in and/or from the NoSQL database functionality or

-- at retrieving configuration information from the NoSQL database functionality, is handled by the following steps:

-- in a third step - after the second step - the further query message is transmitted from the specific network node with the help of a query router module, wherein the query router module routes the further query message to the database entity comprising at least one graph partition of the graph of the further query message,

-- in a fourth step - after the third step - the further query message is received by specific database entity comprising at least one graph partition of the graph of the further query message, wherein, with the help of a query handler module having access to the NoSQL database functionality, the further query message is solved and/or answered.

[0008] According to the present invention, it is thereby advantageously possible to employ a (highly) distributed database functionality (the NoSQL database functionality) for an enhanced management of configuration information and/or stateful information in a communication network, wherein query messages between the plurality of network nodes and the NoSQL database functionality are transmitted using a data modeling language with a graph-like or tree-like data structure. In particular, it is possible that vertices (at least two vertices or more than two vertices) that are referenced in a specific graph of a specific query message are distributed among a group of graph partitions (an initial group of graph partitions and/or a modified group of graph partitions) across different database entities which are part of the NoSQL (Not only SQL, Not only structured query language) database functionality as part of a first step and a second step. Thereby, it is advantageously possible to reduce the number of network hops (between different database entities) needed to solve a query message.

[0009] In case the at least two vertices referenced in the specific graph are distributed within a group of initial graph partitions across the database entities of the NoSQL database functionality, it is advantageously possible that the plurality of query messages is a typical (especially an estimated) set of query messages. By tracking and analyzing such a plurality of query messages in a first step, it is possible that workload information for a specific vertex referenced (or comprised) in the specific graph of the specific query message of the plurality of query messages is generated by a query profile module. It is preferably possible that (respective) workload information for many (or all) vertices referenced (or comprised

in a) specific graph of the specific query message of the plurality of query messages is generated by the query profile module. It is possible that the generated workload information is a measure for the workload (or the amount of required resources) when handling/routing the plurality of query messages associated to storing the specific vertex on a specific database entity (and/or the specific vertex in a specific graph partition on a specific database entity). It is advantageously possible that the workload and/or the amount of resources required for routing/handling/answering the plurality of query messages is minimized because in dependence of the workload information, a partitioning of the specific graph into a group of initial graph partitions (and preferably a partitioning of many graphs into groups of initial graph partitions) is performed by a graph partitioning module, wherein each graph partition of the group of initial graph partitions is distributed to at least one different database entity of the plurality of database entities in a second step.

[0010] Alternatively or additionally it is advantageously possible to optimize/enhance the partitioning in a first step and second step. Therein the at least two vertices referenced in the specific graph are distributed within a group of modified graph partitions across the database entities of the NoSQL database functionality. In this case, it is possible that the plurality of query messages is a typical set of query messages arising during operation of the system (during the exchange of query messages between network nodes and the NoSQL database functionality). By tracking and analyzing such a plurality of query messages in a first step, it is possible that workload information for a specific vertex referenced (or comprised) in the specific graph of the specific query message of the plurality of query messages is generated by a query profile module. It is preferably possible that (respective) workload information for many (or all) vertices referenced (or comprised in a) specific graph of the specific query message of the plurality of query messages is generated by the query profile module. It is possible that the generated workload information is a measure for the workload (or the amount of required resources) when handling/routing the plurality of query messages associated to storing the specific vertex on a specific database entity (and/or the specific vertex in a specific graph partition on a specific database entity). It is advantageously possible that the workload and/or the amount of resources required for routing/handling/answering the plurality of query messages is minimized because in dependence of the workload information, a partitioning of the specific graph into a group of modified graph partitions (and preferably a partitioning of many graphs into groups of modified graph partitions) is performed by a graph partitioning module, wherein each graph partition of the group of modified graph partitions is distributed to at least one different database entity of the plurality of database entities in a second step. Therein, it is especially possible that a group of initial graph partitions is changed, such that the graph partitioning is adapted to the plurality of query messages arising during operation of the system (during the exchange of query messages between network nodes and the NoSQL database functionality).

[0011] According to the present invention, after the partitioning in the second step is done, it is possible that a further query message, transmitted from a specific network node of the plurality of network nodes (and typically many further query messages from many different network nodes), directed at editing/writing/retrieving stateful information in and/or from the NoSQL database functionality or at retrieving configuration information from the NoSQL database functionality is handled in an efficient manner by a third step and a fourth step.

[0012] According to the present invention, it is advantageously possible that a message exchange of query messages using a graph-like or tree-like data structure between a plurality of network nodes and a NoSQL database functionality is handled more efficiently because the vertices referenced in the graphs of the query messages of the plurality of query messages (or at least the specific vertex referenced in the specific graph of the specific query message of the plurality of query messages) are distributed among a plurality of database entities that are part of the NoSQL database functionality in an efficient way such that, when handling a (further) query message, a reduced of amount of database entities of the plurality of database entities is involved in handling the (further) query message. Therein, it is advantageously possible to realize a distributed database functionality for a communication network, wherein network nodes employ query messages with a graph-like or tree-like data structure.

[0013] According to the present invention, it is advantageously possibly that a network-wide configuration store (database functionality) is realized, wherein features such as N-way master-less replication, multi-data-center deployment, handling of consistency versus availability issues in face of network partitions, linear scale-out properties and high throughput are achievable.

[0014] According to an embodiment of the present invention, the query router module is at least partly comprised in or associated with the specific network node.

[0015] It is advantageously possible according to an embodiment of the present invention that the query router module (or parts thereof) is partly comprised in or associated with many (or all) network nodes of the plurality of network nodes.

[0016] According to an embodiment of the present invention, the query profile module is at least partly comprised in or associated with the specific database entity, wherein especially the query profile module is partly comprised in or associated with each database entity of the plurality of database entities, wherein the graph partitioning module is at least partly comprised in or associated with the specific database entity, wherein especially the graph partitioning module is partly comprised in or associated with each database entity of the plurality of database entities, wherein query handler module is at least

partly comprised in or associated with the specific database entity, wherein especially the query handler module is partly comprised in or associated with each database entity of the plurality of database entities.

[0017] It is advantageously possible according to an embodiment of the present invention that the query profile module is partly comprised in or associated with many database entities of the plurality of database entities. It is additionally or alternatively possible that the graph partitioning module is partly comprised in or associated with many database entities of the plurality of database entities. It is additionally or alternatively possible that the query handler module is partly comprised in or associated with many database entities of the plurality of database entities. It is possible that the query profile module, the graph partitioning module and the query handler module (or parts thereof) are bundled together at a database entity, either as local processes communicating over local sockets or as linked-in libraries or extensions of the database functionality application programming interface (API) (e.g., a Cassandra API).

[0018] According to an embodiment of the present invention, the workload information for the specific vertex is further a function of a cost value associated to answering the query messages of the plurality of query messages referencing the specific vertex, wherein especially each query message of the plurality of query messages is weighted with a predefined relative importance value.

[0019] Thereby, it is advantageously possible according to an embodiment of the present invention that the vertices referenced in the graphs of query messages of the plurality of query messages are distributed among a plurality of database entities that are part of the NoSQL database functionality in a further enhanced way.

[0020] According to an embodiment of the present invention, the data modeling language with a graph-like or tree-like data structure is YANG and the NoSQL database functionality is a Cassandra-based NoSQL database functionality.

[0021] Thereby it is possible according to an embodiment of the present invention that a highly scalable and distributed database functionality is employed.

[0022] According to an embodiment of the present invention, the partitioning of the specific graph into the group of initial graph partitions during the second step is performed such that a workload indicated by the workload information is minimized.

[0023] Thereby, it is advantageously possible that the workload and/or the amount of resources required for routing/handling/answering the plurality of query messages is minimized in an enhanced way.

[0024] According to an embodiment of the present invention, in the first step workload information for a further specific vertex referenced in the specific graph is generated analogous to the generation of the workload information for the specific vertex, wherein especially in the first step workload information for each specific vertex referenced in the specific graph is generated analogous to the generation of the workload information for the specific vertex.

[0025] It is possible according to an embodiment of the present invention that in the first step workload information for a plurality of further specific vertices referenced in the specific graph is generated analogous to the generation of the workload information for the specific vertex.

[0026] According to an embodiment of the present invention, in the first step workload information for each vertex referenced in any of the graphs of the plurality of query messages is generated analogous to the generation of the workload information for the specific vertex, wherein the partitioning of each graph of the plurality of query messages into a respective group of initial graph partitions and/or a respective group of modified graph partitions - in the second step - is performed analogous to the partitioning of the specific graph into the group of initial graph partitions and/or the group of modified graph partitions.

[0027] According to an embodiment of the present invention, after the second step either:

-- the specific vertex is comprised in only one initial graph partition or modified graph partition or

-- the specific vertex is comprised in a plurality of initial graph partitions or modified graph partitions.

[0028] Thereby, it is possible according to an embodiment of the present invention that a specific vertex might by comprised in more than one initial graph partition or modified graph partition. Additionally or alternatively, it is possible that a further specific vertex is comprised in only one initial graph partition or modified graph partition.

[0029] According to an embodiment of the present invention, the partitioning in the second step is performed utilizing a basic partition cut, a complementary partition cut and/or redundant partitioning.

[0030] The object of the present invention is also achieved by system for an enhanced management of configuration information and/or stateful information in a communication network, wherein the communication network comprises a plurality of network and a distributed NoSQL database functionality, wherein the distributed NoSQL database functionality comprises a plurality of database entities, wherein query messages between the plurality of network nodes and the NoSQL database functionality are transmittable using a data modeling language with a graph-like or tree-like data structure, wherein the query messages comprise graphs referencing vertices stored in the plurality of database entities, wherein a specific graph of a specific query message comprises references to at least two vertices, wherein the system is configured to distribute the at least two vertices referenced in the specific graph within a group of initial graph partitions across the database entities of the NoSQL database functionality by:

-- a query profile module is configured to track and

analyze a plurality of query, wherein the query profile module is configured to generate workload information for a specific vertex referenced in the specific graph of the specific query message of the plurality of query messages associated to storing the specific vertex on a specific database entity out of the plurality of database entities, wherein the workload information for the specific vertex is a function of

-- the number of query messages of the plurality of query messages referencing the specific vertex,
-- the frequency of query messages of the plurality of query messages referencing the specific vertex,
-- wherein, in case the specific vertex is the first or last vertex of a graph partition, the workload information is additionally a function of a communication cost value associated with hopping/transitioning to a vertex in a different graph partition,

-- in dependence of the workload information, a graph partitioning module is configured to perform a partitioning of the specific graph into a group of initial graph partitions, wherein each graph partition of the group of initial graph partitions is configured to be distributed to at least one different database entity of the plurality of database entities, and/or

wherein the system is configured to distribute the at least two vertices referenced in the specific graph within a group of modified graph partitions across the database of the NoSQL database functionality by:

-- a query profile module is configured to track and analyze a plurality of query, wherein the query profile module is configured to generate workload information for a specific vertex referenced in the specific graph of the specific query message of the plurality of query messages associated to storing the specific vertex on a specific database entity out of the plurality of database entities, wherein the workload information for the specific vertex is a function of

-- the number of query messages of the plurality of query messages referencing the specific vertex,
-- the frequency of query messages of the plurality of query messages referencing the specific vertex,
-- wherein, in case the specific vertex is the first or last vertex of a graph partition, the workload information is additionally a function of a communication cost value associated with hopping/transitioning to a vertex in a different graph partition,

-- in dependence of the workload information, a graph partitioning module is configured to perform a partitioning of the specific graph into a group of modified graph partitions, wherein each graph partition of the group of modified graph partitions is configured to be distributed to at least one different database entity of the plurality of database entities,

wherein the system is configured to handle a further query message, transmittable from a specific network node of the plurality of network nodes and directed either

-- at editing and/or writing and/or retrieving stateful information in and/or from the NoSQL database functionality or
-- at retrieving configuration information from the NoSQL database functionality, by:
-- the further query message is configured to be transmitted from the specific network node with the help of a query router module, wherein the query router module is configured to route the further query message to the database entity comprising at least one graph partition of the graph of the further query message,
-- the further query message is configured to be received by specific database entity comprising at least one graph partition of the graph of the further query message, wherein, with the help of a query handler module having access to the NoSQL database functionality, the further query message is configured to be solved and/or answered.

[0031] Additionally, the object of the present invention is achieved by program comprising a computer readable program code which, when executed in part on a query profile module, a graph partitioning module, a query router module, a query handler module and/or on a database entity and/or on a network node, causes the query profile module, the graph partitioning module, the query router module, the query handler module and/or the database entity and/or the network node to perform a method according to an embodiment of the present invention.

[0032] Additionally, the object of the present invention is achieved by computer program product for an enhanced management of configuration information and/or stateful information in a communication network, the computer program product comprising a computer program stored on a storage medium, the computer program comprising program code which, when executed in part on a query profile module, a graph partitioning module, a query router module, a query handler module and/or on a database entity and/or on a network node, causes the query profile module, the graph partitioning module, the query router module, the query handler module and/or the database entity and/or the network node to perform a method according to an embodiment of the present invention.

[0033] The inventive system, program and computer

program product share the advantages described for the inventive method and embodiments of the inventive method.

[0034] These and other characteristics, features and advantages of the present invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, which illustrate, by way of example, the principles of the invention. The description is given for the sake of example only, without limiting the scope of the invention. The reference figures quoted below refer to the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0035]

Figure 1 schematically illustrates a system according to an embodiment of the present invention.

Figure 2 schematically illustrates a specific network node of the plurality of network nodes according to an embodiment of the present invention.

Figure 3 schematically illustrates a specific database entity of the plurality of database entities according to an embodiment of the present invention.

DETAILED DESCRIPTION

[0036] The present invention will be described with respect to particular embodiments and with reference to certain drawings but the invention is not limited thereto but only by the claims. The drawings described are only schematic and are non-limiting. In the drawings, the size of some of the elements may be exaggerated and not drawn on scale for illustrative purposes.

[0037] Where an indefinite or definite article is used when referring to a singular noun, e.g. "a", "an", "the", this includes a plural of that noun unless something else is specifically stated.

[0038] Furthermore, the terms first, second, third and the like in the description and in the claims are used for distinguishing between similar elements and not necessarily for describing a sequential or chronological order. It is to be understood that the terms so used are interchangeable under appropriate circumstances and that the embodiments of the invention described herein are capable of operation in other sequences than described or illustrated herein.

[0039] In **Figure 1,** a system for an enhanced management of configuration information and/or stateful information in a communication network according to an embodiment of the present invention is schematically illustrated. The system comprises the communication network, wherein the communication network comprises a plurality of network nodes 11, 12, 13 and a distributed NoSal database functionality 20. The distributed NoSal database functionality 20 comprises a plurality of database entities 21, 22, 23, which may or may not be physically separated from each other (e.g., at different locations). Query messages are transmittable between the plurality of network nodes 11, 12, 13 and the NoSal database functionality 20 employing a data modeling language with a graph-like or tree-like data structure, especially YANG. Such query message can, e.g., be directed at editing and/or writing and/or retrieving stateful information in and/or from the NoSal database functionality 20 or at retrieving configuration information from the NoSal database functionality 20. Therein, communication links between the plurality of network node 11, 12, 13 and the plurality of database entities 21, 22, 23 exist. It is also possible that there are communication links between different database entities 21, 22, 23 of the plurality of database entities 21, 22, 23 (not shown) and between different network nodes 11, 12, 13 of the plurality of network nodes 11, 12, 13 (not shown).

[0040] According to an embodiment of the present invention, it is possible that the at least two vertices referenced in the specific graph are distributed within a group of initial graph partitions 400 across/among the database entities 21, 22, 23 of the NoSQL database functionality 20 by a first and second step. In the first step, a plurality of query messages is tracked and analyzed by a query profile module 30, which is preferably partly comprised in each database entity 21, 22, 23, wherein the query profile module 30 generates workload information 300 for a specific vertex referenced in the specific graph of the specific query message of the plurality of query messages associated to storing the specific vertex on a specific database entity 21 out of the plurality of database entities 21, 22, 23. The workload information 300 for the specific vertex is a function of the number of query messages of the plurality of query messages referencing the specific vertex, the frequency of query messages of the plurality of query messages referencing the specific vertex, a cost value associated to answering all query messages of the plurality of query messages referencing the specific vertex and, in case the specific vertex is the first or last vertex of a graph partition 400 (i.e., an edge), the workload information 300 is additionally a function of a communication cost value associated with hopping/transitioning to a vertex in a different graph partition 400. In the second step - after the first step - in dependence of the workload information 300, a partitioning of the specific graph into a group of initial graph partitions 400 is performed by a graph partitioning module 40, which is preferably partly comprised in each database entity 21, 22, 23. Therein, each graph partition of the group of initial graph partitions 400 is distributed to at least one different database entity 21, 22, 23 of the plurality of database entities 21, 22, 23. The distribution is done such that the workload/cost/amount of resources and/or potentially time, associated to handling/routing/answering the plurality of query messages is minimized, by minimizing the workload/cost/amount of resources and/or potentially time indicated by the workload information. The first and

second steps can also be performed for a group of modified graph partitions 400 instead of the group of initial graph partitions 400, as described above. It is thereby advantageously possible according to the present invention to employ a group of query messages in the first and second step that arises during operation of the system, i.e., during normal communication between the plurality of network nodes 11, 12, 13 and the plurality of database entities 21, 22, 23, when distributing the vertices referenced in the specific graph within a group of modified graph partitions 400. Therein by the first and second step, the partitioning is done such that a further query message, transmitted from a specific network node 11 of the plurality of network nodes 11, 12, 13 and directed at editing and/or writing and/or retrieving stateful information in and/or from the NoSQL database functionality 20 or at retrieving configuration information from the NoSQL database functionality 20, is handled by a third and fourth step. In the third step - after the second step - the further query message is transmitted from the specific network node 11 with the help of a query router module 50, which is at least partly comprised in or associated with the specific network 11, wherein the query router module 50 routes the further query message to the database entity 21, 22, 23 comprising at least one graph partition 400 of the graph of the further query message. Therein, it is possible that query router module 50 is aware of the portioning (e.g., by having access to a query routing map 100, in which information on the partitioning of the group of initial graph partitions 400 and/or the group of modified graph partitions 400 is stored/comprised), i.e., is aware of which vertex is stored in which graph partition and where on which database entity 21, 22, 23, the respective graph partition is located. Therein, it is possible that the further query message is directly routed by the query router module 50 to the correct specific database entity 21, 22, 23. In a fourth step - after the third step - the further query message is received by the specific database entity 21, 22, 23, comprising at least one graph partition 400 of the graph of the further query message, wherein the further query message is especially received by a query handler module 60, which partly comprised in or associated with the specific database entity 21, 22, 23. With the help of a query handler module 60 having access to the NoSQL database functionality 20, the further query message is solved and/or answered.

[0041] In **Figure 2,** a specific network node 11 of the plurality of network nodes 11, 12, 13 according to an embodiment of the present invention is schematically illustrated. The specific network node 11 partly comprises a query router module 50 (at least a part thereof). The query router module 50 (or the part of the query router module 50) is configured to route and or transmit a further query message from the specific network node 11 to the database entity 21, 22, 23 of the NoSQL database functionality 20 comprising at least one graph partition 400 of the graph of the further query message, as depicted by the arrow starting at the query router module 50 in Figure 2. Therein, it is possible that the query router module 50 routes/transmits any particular query message from the specific network node 11 to a database entity 21, 22, 23, which comprises a graph partition 400, comprising a vertex referenced in the graph of the particular query message. Furthermore, it is preferred that the specific network node 11 and/or the query router module 50 comprises or has access to a query routing map 100 (or a part thereof). Other network nodes 12, 13 of network nodes 11, 12, 13 can be formed similarly or identically to the specific network node 11.

[0042] In **Figure 3,** a specific database entity 21 of the plurality of database entities 21, 22, 23 according to an embodiment of the present invention is schematically illustrated. A query profile module 30 is at least partly comprised in or associated with the specific database entity 21. Furthermore, a graph partitioning module 40 is at least partly comprised in or associated with the specific database entity 21. Also, a query handler module 60 is at least partly comprised in or associated with the specific database entity 21. An initial graph partition 400 or a modified graph partition 400 (or many initial or modified graph partitions 400) is comprised in the specific database entity 21. The initial graph partition 400 or the modified graph partition 400 is a result of the partitioning of a specific graph of a specific query message of a plurality of query messages or of many specific graphs of query messages of a plurality of query messages the by the graph partitioning module 40 in a second step. Furthermore, workload information 300 is comprised in the specific database entity 21. The workload information 300 is generated by the query profile module 30, after the query profile module tracked and analyzed a plurality of query messages. The workload information 300 concerns a specific vertex referenced in the specific graph of the specific query message of the plurality of query messages and is associated to storing the specific vertex on a specific database entity 21 out of the plurality of database entities 21, 22, 23. In particular, the workload information 300 is a function of the number of query messages of the plurality of query messages referencing the specific vertex, the frequency of query messages of the plurality of query messages referencing the specific vertex and - in case the specific vertex is the first or last vertex of a graph partition 400 - , the workload information 300 is additionally a function of a communication cost value associated with hopping/transitioning to a vertex in a different graph partition 400. It is possible that workload information 300 concerning each vertex referenced in the specific graph of the specific query message is comprised in the specific database entity 21 and also that workload information 300 concerning each vertex referenced in each graph each query message of the plurality of query message is comprised in the specific database entity 21 (especially at least for every vertex that is comprised in the specific database entity 21). Other database entities 22, 23 of the plurality of database entities 21, 22, 23 can be formed similarly or identically to the specific database entities 21.

Claims

1. Method for an enhanced management of configuration information and/or stateful information in a communication network, wherein the communication network comprises a plurality of network nodes (11, 12, 13) and a distributed NoSQL database functionality (20), wherein the distributed NoSQL database functionality (20) comprises a plurality of database entities (21, 22, 23), wherein query messages between the plurality of network nodes (11, 12, 13) and the NoSQL database functionality (20) are transmitted using a data modeling language with a graph-like or tree-like data structure, wherein the query messages comprise graphs referencing vertices stored in the plurality of database entities (21, 22, 23), wherein a specific graph of a specific query message comprises references to at least two vertices, wherein the at least two vertices referenced in the specific graph are distributed within a group of initial graph partitions (400) across the database entities (21, 22, 23) of the NoSQL database functionality (20) by the following steps:

    -- in a first step, a plurality of query messages is tracked and analyzed by a query profile module (30), wherein the query profile module (30) generates workload information (300) for a specific vertex referenced in the specific graph of the specific query message of the plurality of query messages associated to storing the specific vertex on a specific database entity (21) out of the plurality of database entities (21, 22, 23), wherein the workload information (300) for the specific vertex is a function of

        -- the number of query messages of the plurality of query messages referencing the specific vertex,
        -- the frequency of query messages of the plurality of query messages referencing the specific vertex,
        -- wherein, in case the specific vertex is the first or last vertex of a graph partition (400), the workload information (300) is additionally a function of a communication cost value associated with hopping/transitioning to a vertex in a different graph partition (400),

    -- in a second step - after the first step - in dependence of the workload information (300), a partitioning of the specific graph into a group of initial graph partitions (400) is performed by a graph partitioning module (40), wherein each graph partition of the group of initial graph partitions (400) is distributed to at least one different database entity (21, 22, 23) of the plurality of database entities (21, 22, 23),

    and/or

    wherein the at least two vertices referenced in the specific graph are distributed within a group of modified graph partitions (400) across the database entities (21, 22, 23) of the NoSQL database functionality (20) by the following steps:

    -- in a first step, a plurality of query messages is tracked and analyzed by a query profile module (30), wherein the query profile module (30) generates workload information (300) for a specific vertex referenced in the specific graph of the specific query message of the plurality of query messages associated to storing the specific vertex on a specific database entity (21) out of the plurality of database entities (21, 22, 23), wherein the workload information (300) for the specific vertex is a function of

        -- the number of query messages of the plurality of query messages referencing the specific vertex,
        -- the frequency of query messages of the plurality of query messages referencing the specific vertex,
        -- wherein, in case the specific vertex is the first or last vertex of a graph partition (400), the workload information (300) is additionally a function of a communication cost value associated with hopping/transitioning to a vertex in a different graph partition (400),

    -- in a second step - after the first step - in dependence of the workload information (300), a partitioning of the specific graph into a group of modified graph partitions (400) is performed by a graph partitioning module (40), wherein each graph partition (400) of the group of modified graph partitions (400) is distributed to at least one different database entity (21, 22, 23) of the plurality of database entities (21, 22, 23),

    wherein a further query message, transmitted from a specific network node(11) of the plurality of network nodes (11, 12, 13) and directed either

    -- at editing and/or writing and/or retrieving stateful information in and/or from the NoSQL database functionality (20) or
    -- at retrieving configuration information from the NoSQL database functionality (20), is handled by the following steps:
    -- in a third step - after the second step - the further query message is transmitted from the specific network node (11) with the help of a query router module (50), wherein the query router module (50) routes the further query message to the database entity (21, 22, 23) comprising at

least one graph partition (400) of the graph of the further query message,
-- in a fourth step - after the third step - the further query message is received by specific database entity (21, 22, 23) comprising at least one graph partition (400) of the graph of the further query message, wherein, with the help of a query handler module (60) having access to the NoSQL database functionality (20), the further query message is solved and/or answered.

2. Method according to claim 1, wherein the query router module (50) is at least partly comprised in or associated with the specific network node (11).

3. Method according to any of the preceding claims, wherein the query profile module (30) is at least partly comprised in or associated with the specific database entity (21), wherein especially the query profile module (30) is partly comprised in or associated with each database entity (21, 22, 23) of the plurality of database entities (21, 22, 23), wherein the graph partitioning module (40) is at least partly comprised in or associated with the specific database entity (21), wherein especially the graph partitioning module (40) is partly comprised in or associated with each database entity (21, 22, 23) of the plurality of database entities (21, 22, 23), wherein query handler module (60) is at least partly comprised in or associated with the specific database entity (21), wherein especially the query handler module (60) is partly comprised in or associated with each database entity (21, 22, 23) of the plurality of database entities (21, 22, 23).

4. Method according to any of the preceding claims, wherein the workload information (300) for the specific vertex is further a function of a cost value associated to answering the query messages of the plurality of query messages referencing the specific vertex, wherein especially each query message of the plurality of query messages is weighted with a predefined relative importance value.

5. Method according to any of the preceding claims, wherein the data modeling language with a graph-like or tree-like data structure is YANG and the NoSQL database functionality is a Cassandra-based NoSQL database functionality.

6. Method according to any of the preceding claims, wherein the partitioning of the specific graph into the group of initial graph partitions (400) during the second step is performed such that a workload indicated by the workload information (300) is minimized.

7. Method according to any of the preceding claims, wherein in the first step workload information (300) for a further specific vertex referenced in the specific graph is generated analogous to the generation of the workload information (300) for the specific vertex, wherein especially in the first step workload information (300) for each specific vertex referenced in the specific graph is generated analogous to the generation of the workload information (300) for the specific vertex.

8. Method according to any of the preceding claims, wherein in the first step workload information (300) for each vertex referenced in any of the graphs of the plurality of query messages is generated analogous to the generation of the workload information (300) for the specific vertex, wherein the partitioning of each graph of the plurality of query messages into a respective group of initial graph partitions (400) and/or a respective group of modified graph partitions (400) - in the second step - is performed analogous to the partitioning of the specific graph into the group of initial graph partitions (400) and/or the group of modified graph partitions (400).

9. Method according to any of the preceding claims, wherein after the second step either:

-- the specific vertex is comprised in only one initial graph partition (400) or modified graph partition (400) or
-- the specific vertex is comprised in a plurality of initial graph partitions (400) or modified graph partitions (400).

10. Method according to any of the preceding claims, wherein the partitioning in the second step is performed utilizing a basic partition cut, a complementary partition cut and/or redundant partitioning.

11. System for an enhanced management of configuration information and/or stateful information in a communication network, wherein the communication network comprises a plurality of network nodes (11, 12, 13) and a distributed NoSQL database functionality (20), wherein the distributed NoSQL database functionality (20) comprises a plurality of database entities (21, 22, 23), wherein query messages between the plurality of network nodes (11, 12, 13) and the NoSQL database functionality (20) are transmittable using a data modeling language with a graph-like or tree-like data structure, wherein the query messages comprise graphs referencing vertices stored in the plurality of database entities (21, 22, 23), wherein a specific graph of a specific query message comprises references to at least two vertices, wherein the system is configured to distribute the at least two vertices referenced in the specific graph within a group of initial graph partitions (400) across the database entities (21, 22, 23) of the NoSQL da-

tabase functionality (20) by:

-- a query profile module (30) is configured to track and analyze a plurality of query, wherein the query profile module (30) is configured to generate workload information (300) for a specific vertex referenced in the specific graph of the specific query message of the plurality of query messages associated to storing the specific vertex on a specific database entity (21) out of the plurality of database entities (21, 22, 23), wherein the workload information (300) for the specific vertex is a function of

-- the number of query messages of the plurality of query messages referencing the specific vertex,
-- the frequency of query messages of the plurality of query messages referencing the specific vertex,
-- wherein, in case the specific vertex is the first or last vertex of a graph partition (400), the workload information (300) is additionally a function of a communication cost value associated with hopping/transitioning to a vertex in a different graph partition (400),

-- in dependence of the workload information (300), a graph partitioning module (40) is configured to perform a partitioning of the specific graph into a group of initial graph partitions (400), wherein each graph partition (400) of the group of initial graph partitions (400) is configured to be distributed to at least one different database entity (21, 22, 23) of the plurality of database entities (21, 22, 23),

and/or
wherein the system is configured to distribute the at least two vertices referenced in the specific graph within a group of modified graph partitions (400) across the database entities (21, 22, 23) of the NoSQL database functionality (20) by:

-- a query profile module (30) is configured to track and analyze a plurality of query, wherein the query profile module (30) is configured to generate workload information (300) for a specific vertex referenced in the specific graph of the specific query message of the plurality of query messages associated to storing the specific vertex on a specific database entity (21) out of the plurality of database entities (21, 22, 23), wherein the workload information (300) for the specific vertex is a function of

-- the number of query messages of the plurality of query messages referencing the specific vertex,
-- the frequency of query messages of the plurality of query messages referencing the specific vertex,
-- wherein, in case the specific vertex is the first or last vertex of a graph partition (400), the workload information (300) is additionally a function of a communication cost value associated with hopping/transitioning to a vertex in a different graph partition (400),

-- in dependence of the workload information (300), a graph partitioning module (40) is configured to perform a partitioning of the specific graph into a group of modified graph partitions (400), wherein each graph partition (400) of the group of modified graph partitions (400) is configured to be distributed to at least one different database entity (21, 22, 23) of the plurality of database entities (21, 22, 23), wherein the system is configured to handle a further query message, transmittable from a specific network node(11) of the plurality of network nodes (11, 12, 13) and directed either
-- at editing and/or writing and/or retrieving stateful information in and/or from the NoSQL database functionality (20) or
-- at retrieving configuration information from the NoSQL database functionality (20), by:
-- the further query message is configured to be transmitted from the specific network node (11) with the help of a query router module (50), wherein the query router module (50) is configured to route the further query message to the database entity (21, 22, 23) comprising at least one graph partition (400) of the graph of the further query message,
-- the further query message is configured to be received by specific database entity (21, 22, 23) comprising at least one graph partition (400) of the graph of the further query message, wherein, with the help of a query handler module (60) having access to the NoSQL database functionality (20), the further query message is configured to be solved and/or answered.

12. Program comprising a computer readable program code which, when executed in part on a query profile module (30), a graph partitioning module (40), a query router module (50), a query handler module (60) and/or on a database entity and/or on a network node, causes the query profile module (30), the graph partitioning module (40), the query router module (50), the query handler module (60) and/or the database entity and/or the network node to perform a method according to any one of claims 1 to 10.

13. Computer program product for an enhanced man-

agement of configuration information and/or stateful information in a communication network, the computer program product comprising a computer program stored on a storage medium, the computer program comprising program code which, when executed in part on a query profile module (30), a graph partitioning module (40), a query router module (50), a query handler module (60) and/or on a database entity and/or on a network node, causes the query profile module (30), the graph partitioning module (40), the query router module (50), the query handler module (60) and/or the database entity and/or the network node to perform a method according to any one of claims 1 to 10.

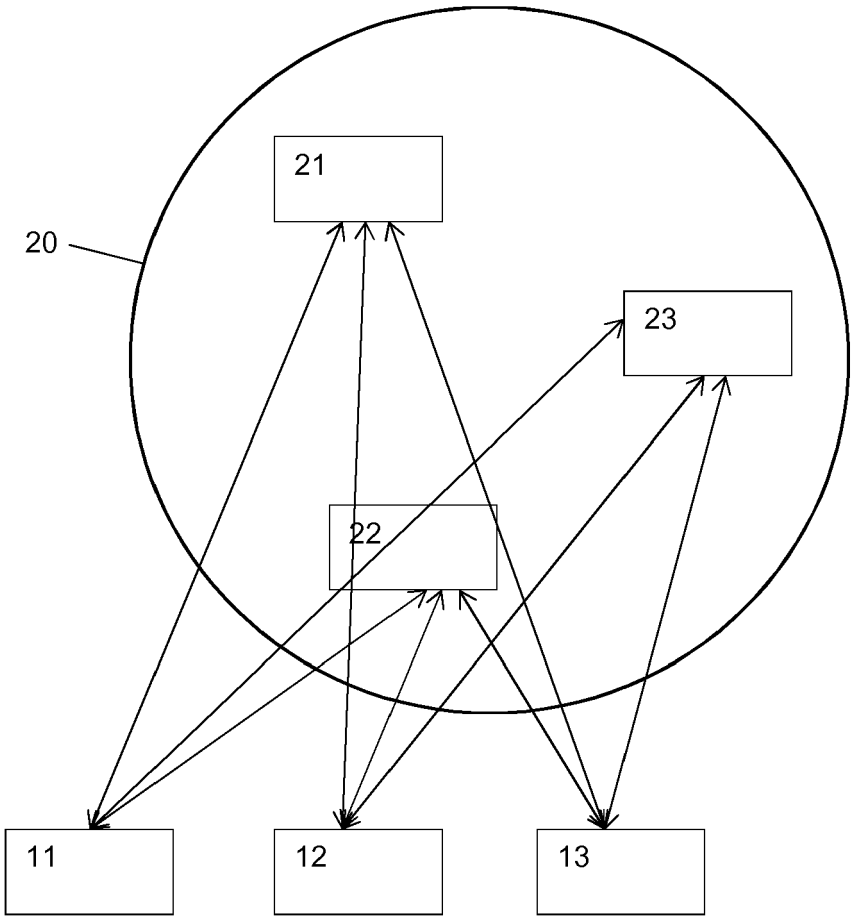

Fig. 1

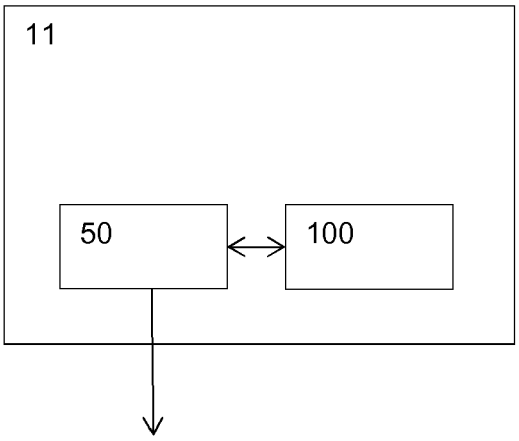

Fig. 2

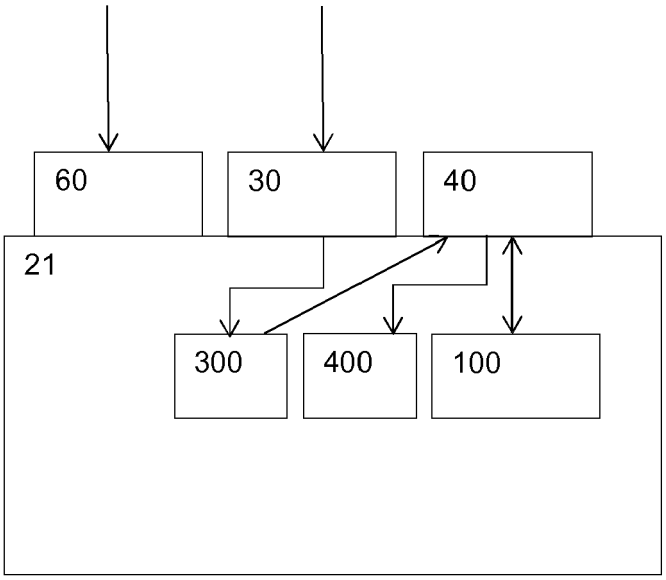

Fig. 3

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 17 17 8243

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	US 9 031 994 B1 (CAO YU [CN] ET AL) 12 May 2015 (2015-05-12) * abstract; figures 1,7 * * column 5, line 54 - column 6, line 25 * * column 11, line 4 - line 50 * -----	1-13	INV. G06F17/30
A	US 2014/114910 A1 (CAO YU [CN] ET AL) 24 April 2014 (2014-04-24) * figures 1-2 * * paragraph [0032] - paragraph [0039] * -----	1-13	

TECHNICAL FIELDS SEARCHED (IPC)

G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Berlin	6 December 2017	Nazzaro, Antonio

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 17 17 8243

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

06-12-2017

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 9031994 B1	12-05-2015	NONE	
US 2014114910 A1	24-04-2014	CN 103714098 A	09-04-2014
		US 2014114910 A1	24-04-2014

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82